Semantic Role Labeling of Speech Transcripts

Niraj Shrestha, Ivan Vulić, and Marie-Francine Moens

Department of Computer Science, KU Leuven, Leuven, Belgium
{niraj.shrestha,ivan.vulic,marie-francine.moens}@cs.kuleuven.be

Abstract. Speech data has been established as an extremely rich and important source of information. However, we still lack suitable methods for the semantic annotation of speech that has been transcribed by automated speech recognition (ASR) systems . For instance, the semantic role labeling (SRL) task for ASR data is still an unsolved problem, and the achieved results are significantly lower than with regular text data. SRL for ASR data is a difficult and complex task due to the absence of sentence boundaries, punctuation, grammar errors, words that are wrongly transcribed, and word deletions and insertions. In this paper we propose a novel approach to SRL for ASR data based on the following idea: (1) combine evidence from different segmentations of the ASR data, (2) jointly select a good segmentation, (3) label it with the semantics of PropBank roles. Experiments with the OntoNotes corpus show improvements compared to the state-of-the-art SRL systems on the ASR data. As an additional contribution, we semi-automatically align the predicates found in the ASR data with the predicates in the gold standard data of OntoNotes which is a quite difficult and challenging task, but the result can serve as gold standard alignments for future research.

Keywords: Semantic role labeling, speech data, ProBank, OntoNotes.

1 Introduction

Semantic role labeling (SRL) regards the process of predicting the predicate argument structure of a natural language utterance by detecting the predicate and by detecting and classifying the arguments of the predicate according to their underlying semantic role. SRL reveals more information about the content than a syntactic analysis in the field of natural language processing (NLP) in order to better understand "who" did "what" to "whom", and "how", "when" and "where". For example, in the following two sentences:

Mary opened the door.
The door was opened by Mary.

Syntactically, the subjects and objects are different. "Mary" and "the door" are subject and object in the first sentence respectively, while their syntactic role is swapped in the second sentence. Semantically, in both sentences "Mary" is ARG0 and "the door" is ARG1, since Mary opened the door.

SRL has many key applications in NLP, such as question answering, machine translation, and dialogue systems. Many effective SRL systems have been developed to work with written data. However, when applying popular SRL systems such as ASSERT [1],

A. Gelbukh (Ed.): CICLing 2015, Part II, LNCS 9042, pp. 583–595, 2015.
DOI: 10.1007/978-3-319-18117-2_43

Lund University SRL [2], SWIRL [3], and Illinois SRL [4] on transcribed speech, which was processed by an automatic speech recognizer (ASR), many errors are made due to the specific nature of the ASR-transcribed data.

When a state-of-the art SRL system is applied to ASR data, its performance changes drastically [5] due to many automatic transcription errors such as the lack of sentence boundaries and punctuation, spelling mistakes and insertions, or deletions of words and phrases. The lack of sentence boundaries is another major problem. If a sentence boundary detection system correctly identifies sentence boundaries in the ASR data then the SRL system might produce acceptable results, but unfortunately correct sentence boundary detection in ASR data remains a difficult and error-prone task. In this paper, we investigate whether a correct sentence boundary detector is actually needed for SRL, and whether the recognition of a predicate and its semantic role arguments within a certain window of words would not be sufficient to recover the semantic frames in speech data.

Therefore, we focus on frame segmentation rather than sentence segmentation. A segment is named a *frame segment* when the system finds a predicate and its semantic roles. Frame segments from the ASR data are generated as follows. Taking a fixed window size of words, we generate all possible segments by moving the window slider by one word. Considering this segment as a *pseudo-sentence*, we apply an SRL system, which generates many possible combinations of arguments for a predicate since the same predicate may appear in multiple segments. The system finally chooses the best arguments for the predicate. In summary, in this paper we propose a novel approach to SRL for ASR data based on the following idea:

1. Combine the evidence from different segmentations of the ASR data;
2. Jointly select a good frame segmentation;
3. Label it with the semantics of PropBank roles;

Experiments with the OntoNotes corpus [6] show improvements compared to the state-of-the-art SRL systems on the ASR data. We are able to improve 4.5% and 1.69% in recall and F_1 measure respectively in predicate and semantic role pair evaluation compared to a state-of-the-art semantic role labeling system on the same speech/ASR data set [5]. Our novel approach to SRL for the ASR data is very promising, as it opens plenty of possibilities towards improving the frame detection in speech data without sentence boundary detection. As an additional contribution, we semi-automatically align the predicates found in the ASR data with the predicates in the gold standard data of OntoNotes which is a quite difficult and challenging task, but the result can serve as gold standard alignments for future research.

The following sections first review prior work, then describe the methodology of our approach and the experimental setup, and finally present our evaluation procedure and discuss the results.

2 Prior Work

Semantic role labeling or the task of recognizing basic semantic roles of sentence constituents is a well-established task in natural language processing [7, 8], due to the existence of annotated corpora such as PropBank [9], NomBank [10], FrameNet [11] and shared tasks (CoNLL). Current semantic role labeling systems (e.g., SWIRL: [3],

ASSERT: [1], Illinois SRL: [4], Lund University SRL: [2]) perform well if the model is applied on texts from domains similar to domains of the documents on which the model was trained. Performance for English on a standard collection such as the CoNLL dataset reaches F_1 scores higher than 85% [12] for supervised systems that rely on automatic linguistic processing up to the syntactic level.

On the other hand, semantic role labeling of (transcribed) speech data is very limited, perhaps due to the non-availability of benchmarking corpora. Nevertheless, several authors have stressed the importance of semantic role labeling of speech data, for instance, in the frame of question answering speech interfaces (e.g., [13,14]), speech understanding by robots (e.g., [15]), and speech understanding in general [16]. Favre [5] developed a system for joint dependency parsing and SRL of transcribed speech data in order to be able to handle speech recognition output with word errors and sentence segmentation errors. He uses a classifier for segmenting the sentences trained on sentence-spit ASR data taking into account sentence parse information, lexical features and pause duration. This work is used as a baseline system for our experiments. The performance of semantic role labellers drops significantly (F_1 scores decrease to 50.76% when applying the ASSERT SRL system on ASR data) due to the issues with transcribed speech discussed in introduction. A similar decrease in performance is also noticed when performing SRL on non-well formed texts such as tweets [17].

We hope that this paper will stir up interest of the research community in semantic processing of speech.

3 Methodology

The main objective of this work is to identify suitable ASR segments that represent a predicate with its semantic roles, in a task that we call *frame segmentation*. Frame segments from the ASR data are generated by taking a window of a fixed size, and moving it word-by-word. This way, all possible combinations of segments in which a predicate might appear are generated. Considering each segment as a (pseudo-)sentence, the SRL system generates many possible combinations of arguments for a predicate. Our system then chooses the best arguments for the predicate based on an *evidence-combining approach*. Figure 1 shows a snippet of the raw ASR data, while figure 2 shows the results after moving the fixed-size window word-by-word (brute force segments). After applying the SRL system on these brute force segments, we obtain the labels as shown in figure 3. It is clearly visible that the same predicate occurs in different segments, and also different argument types occur in different segments with different text spans.

To evaluate our approach, we use the OntoNotes corpus [6] annotated with gold standard ProbBank semantic roles [9] and its transcribed speech data.[1] The speech corpus is plain text without any information about time and pause duration. Each token in the corpus is given in its own line with an empty line serving as the sentence boundary mark. We have decided to convert the data into the original raw format which corresponds to the actual ASR output (i.e., no sentence boundary, punctuation marks) by merging all tokens into a single line. The final input corpus then resembles the format of the snippet from figure 1.

[1] The transcribed corpus is provided by [5] with the consent of SRI (`http://www.sri.com`).

a much better looking newsnight i might add as powerless on sits in for anderson and they're and they're both off of that that is not a replacement colin powell thank you for your faith larry and thank you for your graciousness will give and we're going to get started here good evening everybody welcome to newsnight as larry just told you i'm paulus on filling in for the two men anderson cooper and aaron brown he lost his life long ago but there's still something modern science can give him back his identity the mystery of the frozen ever and continues next up the lab and anger in the hood over a sign of the times has

Fig. 1. An example of raw ASR-transcribed speech data

a much better looking newsnight i might add as powerless on sits in for anderson
much better looking newsnight i might add as powerless on sits in for anderson and
better looking newsnight i might add as powerless on sits in for anderson and they
... ...
... ...
and continues next up the lab and anger in the hood over a sign of
continues next up the lab and anger in the hood over a sign of the
next up the lab and anger in the hood over a sign of the times
up the lab and anger in the hood over a sign of the times has

Fig. 2. Brute force segments of window size 15 generated from the raw ASR data

3.1 SRL on Sentence-Segmented ASR Data (Baseline Model)

We compare against a competitive baseline and state-of-the-art model from [5]. We use the same corpus as in [5] which is derived from OntoNotes and which is ASR-transcribed. For our baselinewe use the sentence boundaries as defined in [5]. An SRL system is then applied on the sentences provided in this corpus.

3.2 Longest and Shortest Text Span Selection for Arguments

For a given predicate, there might exist many possible arguments with different argument text spans (see figure 3 again). The first task is to select the optimal text span for

51: good evening everybody welcome to newsnight as larry just told you i 'm paulus on [TARGET filling]
52: evening everybody welcome to newsnight as larry just told you i 'm paulus on [TARGET filling] in
53: everybody welcome to newsnight as larry just told you i 'm paulus on [TARGET filling] in for
54: welcome to newsnight as larry just told you i 'm paulus on [TARGET filling] in [ARG1 for the]
55: to newsnight as larry just told you i 'm paulus on [TARGET filling] in [ARG2 for the two]
56: newsnight as larry just told you i 'm paulus on [TARGET filling] in [ARG1 for the two men]
57: as larry just told you i 'm paulus on [TARGET filling] in [ARG1 for the two men] anderson
58: larry just told you i 'm paulus on [TARGET filling] in [ARG2 for the two men] [ARGM-TMP anderson cooper]
59: just told you i 'm paulus on [TARGET filling] in [ARG1 for the two men] anderson cooper and
60: told you i 'm paulus on [TARGET filling] in [ARG1 for the two men anderson cooper and aaron]
61: you i 'm paulus on [TARGET filling] in [ARG2 for the two men] [ARG1 anderson cooper and aaron brown]
62: i 'm paulus on [TARGET filling] in [ARG2 for the two men] [ARG1 anderson cooper and aaron brown he]
63: paulus on [TARGET filling] in [ARG1 for the two men] anderson cooper and aaron brown he lost
64: on [TARGET filling] in [ARG1 for the two men] anderson cooper and aaron brown he lost his

Fig. 3. Output of the SRL system on brute force segments

each argument. There might occur cases when the text spans of an argument may subsume each other, then either the longest or the shortest text span is chosen. For example, as shown in figure 3, argument type ARG1 exhibits different text spans, ranging from the shortest text span *for the* to the longest span *for the two men anderson cooper and aaron*. In addition, text spans of an argument might differ, and those text spans may not subsume each other. The text span is then selected based on the majority counts according to the occurrence of the two text spans in the corresponding segments, since a predicate cannot have two same argument types for the same dependent text span. Furthermore, text spans of different arguments may subsume each other. If that is the case, the longest or shortest text spans are selected.

Let us assume that the text spans for an argument are as follows:
$w_1 w_2 w_3 \ldots w_{i-1} w_i$
$w_1 w_2 w_3 \ldots w_{i-1} w_i w_{i+1} \ldots w_{j-1} w_j$
$w_1 w_2 w_3 \ldots w_{i-1} w_i w_{i+1} \ldots w_{j-1} w_j w_{j+1} \ldots w_{k-1} w_k$
In the *take-longest* span selection approach, text span $w_1 w_2 w_3 \ldots w_{i-1} w_i w_{i+1} \ldots w_{j-1} w_j w_{j+1} \ldots w_{k-1} w_k$ is chosen. In the *take-shortest* approach text span $w_1 w_2 w_3 \ldots w_{i-1} w_i$ is chosen. There could be also the case where the argument type might have other text spans besides the above ones. Let us assume that there are additional two text spans:
$w_l w_{l+1} w_{l+2} \ldots w_m$
$w_l w_{l+1} w_{l+2} \ldots w_m w_{m+1} w_{m+2} \ldots w_{n-1} w_n$, where $l > k$ or $l < 1$.
Now, with the take-longest selection approach, we have two possible text spans: $w_1 w_2 w_3 \ldots w_{i-1} w_i w_{i+1} \ldots w_{j-1} w_j w_{j+1} \ldots w_{k-1} w_k$ and $w_l w_{l+1} w_{l+2} ... w_m w_{m+1} w_{m+2} \ldots w_n$. Since the argument type can have only one text span, we then choose the first one since the text span $w_1 w_2 w_3 \ldots w_{i-1} w_i$ occurs more times (3 times) than $w_l w_{l+1} w_{l+2} \ldots w_m$ (2 times). The same heuristic is applied in the take-shortest selection approach. We label the take-longest selection approach as **win-n-L**, and the take-shortest approach as **win-n-S**, where the middle '**n**' represents the chosen window size.

3.3 Generating New Segments for a Predicate

Now, we explain a two-pass approach to generating new segments for a predicate. First, we use the output from the SRL system and the brute force approach discussed in 3.2 to detect the predicate. Following that, given this predicate, we identify new segments for the predicate and then again apply the SRL system. In this approach, the SRL system is applied on the brute force segments as discussed above. A predicate might appear in a sequence of segments. We select the first and the last segment of this sequence. These two segments are then merged using two different heuristics to generate two types of new segments. In the first approach, we simply merge the two segments by retaining overlapping tokens. We label this model as **newSeg-V1-win-n**.[2] In the second approach, the new segment starts from the first occurrence of a semantic role argument and ends at the last occurrence of the argument. This model is labeled as **newSeg-V2-win-n** Following that, we remove all the predicate and argument labels and re-run the SRL again on these two new segments.

[2] 'n' in each model is the chosen window size.

For example, given are the following two segments (the first and the last):
First segment: $w_1 w_2 [w_3] \ldots [w_{i-1} w_i] w_{i+1} \ldots w_{j-1} [w_j] w_{j+1} \ldots [w_{k-1} w_k]$
Second segment: $[w_{k-1} w_k] w_{k+1} \ldots w_{l-1} [w_l w_{l-1}] [w_{l+1}] \ldots w_m$
where [] represents argument or predicate labels and tokens inside [] are argument or predicate values. When we generate a new segment with the first approach, we obtain the new segment as:
$w_1 w_2 [w_3] \ldots [w_{i-1} w_i] w_{i+1} \ldots w_{j-1} [w_j] w_{j+1} \ldots [w_{k-1} w_k] w_{k+1} \ldots w_{l-1} [w_l w_{l-1}]$
$[w_{l+1}] \ldots w_m$
After removing the labels, the segment is:
$w_1 w_2 w_3 \ldots w_{i-1} w_i w_{i+1} \ldots w_{j-1} w_j w_{j+1} \ldots w_{k-1} w_k w_{k+1} \ldots w_{l-1} w_l w_{l-1} w_{l+1} \ldots$
w_m
Using the second heuristic, we obtain the new segment as:
$[w_3] \ldots [w_{i-1} w_i] w_{i+1} \ldots w_{j-1} [w_j] w_{j+1} \ldots [w_{k-1} w_k] w_{k+1} \ldots w_{l-1} [w_l w_{l-1}] [w_{l+1}]$
After removing the labels, the new segment is:
$w_3 \ldots w_{i-1} w_i w_{i+1} \ldots w_{j-1} w_j w_{j+1} \ldots w_{k-1} w_k w_{k+1} \ldots w_{l-1} w_l w_{l-1} w_{l+1}$

4 Experimental Setup

We use the OntoNotes release 3 dataset which covers English broadcast and conversation news [6]. The data is annotated in constituent form.

We retain all the settings as described and used by [5]. We use the ASSERT SRL tool [1] for semantic role labeling which was one of the best SRL tools in the CoNLL-2005 SRL shared task[3] and also it outputs semantic roles in constituent form. We only use the sentence boundaries in the baseline state-of-the-art model. We investigate the performance of SRL on different windows sizes to predict the predicates with its semantic role arguments.

4.1 Evaluation

We evaluate different aspects related to the SRL process of identifying predicates and their arguments in the ASR data. Currently, we focus on the ability of the proposed system to predict predicates and arguments, and therefore evaluate the following aspects:

- Evaluation of predicates (**predEval**): We evaluate the ability of the system to predict solely the predicates. Here, we consider only a predicate which has arguments and we adhere to the same strategy in all subsequent evaluation settings. We evaluate the predicates considering different windows sizes. After the predicate identification step, we use these predicates in the subsequent evaluations.
- Evaluation of each argument type, that is, a semantic role (**argEval**): Here, we evaluate each argument type regardless of the found predicate. This evaluation is useful to identify different named entities like time, location, manner, etc.
- Evaluation of a predicate with each argument type (**predArgEval**): Here, we evaluate each argument type in relation to its found predicate. We are evaluating a pair comprising a predicate and its argument type.

[3] `http://www.cs.upc.edu/~srlconll/st05/st05.html`

- Evaluation of a predicate with all argument types (**predArgFullFrameEval**): In this evaluation, we evaluate the full frame for a predicate, that is, the predicate with all its argument types found in a segment. This evaluation is the most severe: if a predicate misses any of its arguments or classifies it wrongly, then the full frame is respectively not recognised or wrongly recognised.
- Evaluation of predicate-argument pairs, but now given a gold labeled correct predicate (**corrPredArgEval**): Here, we evaluate how well the system performs in identifying the correct arguments starting from a correct predicate.

4.2 Predicate Alignment between Gold Standard and Speech Data

Since the speech data does not contain any sentence boundaries and no alignment of its predicates with gold standard, it is necessary to align predicates between the gold standard and predicates identified by the system in speech data for the evaluation. If a predicate occurs once in both corpora, they are aligned using the one-to-one principle. But if a predicate appears more than once in either corpus then we have to align the predicates between two corpora. A predicate from speech data is aligned with a predicate in the gold standard, if three left and three right context words match. If the context words do not match, then we align the predicates manually. There are 41628 predicates in total contained in the gold standard, and the system has identified 25149 predicates out of which 13503 predicates have been aligned in the one-to-one fashion, 5447 predicates have been aligned relying on the left and right context matching, and we have manually aligned the remaining 6199 predicates. This constitutes our predicate alignment between the speech data and the gold standard.

4.3 Evaluation Metrics

Let FL be the final list retrieved by our system, and GL the complete ground truth list. To evaluate different evaluation aspects, we use standard precision (P), recall (R) and F_1 scores for evaluation.

$$P = \frac{|FL \cap GL|}{|FL|} \quad R = \frac{|FL \cap GL|}{|GL|} \quad F_1 = 2 \cdot \frac{P \cdot R}{P + R}$$

We also evaluate the overall system performance in terms of micro-averages and macro-averages for precision, recall and F_1. Suppose we have z arguments types. We then define the evaluation criteria as follows:

$$Micro\_avg(P) = \frac{\sum_{b=1}^{z} |FL \cap GL|}{\sum_{b=1}^{z} |FL|} \quad Micro\_avg(R) = \frac{\sum_{b=1}^{z} |FL \cap GL|}{\sum_{b=1}^{z} |GL|}$$

$$Micro\_avg(F_1) = \frac{2 \times Micro\_avg(P) \times Micro\_avg(R)}{Micro\_avg(P) + Micro\_avg(R)}$$

$$Macro\_avg(P) = \frac{1}{z} \sum_{b=1}^{z} P \quad Macro\_avg(R) = \frac{1}{z} \sum_{b=1}^{z} R$$

$$Macro\_avg(F_1) = \frac{2 \times Macro\_avg(P) \times Macro\_avg(R)}{Macro\_avg(P) + Macro\_avg(R)}$$

5 Results and Discussion

We have investigated the effects of the window size in identifying predicates and its semantic roles. The model **win-20-L** outperforms other variants of our system on a validation set, and also outperforms the baseline system in terms of F_1 measure by 0.94%, while recall is improved by 4.32%. In all models, including the baseline system, recall is lower than precision, and we have noticed that the SRL system is not able to identify the auxiliary verbs like *am, is, are, 're, 'm, has, have, etc.*, which occur many times in the test data. However, they are labeled as predicates with arguments in OntoNotes.

We use the predicates identified by **win-20-L** in other evaluation protocols as already hinted in 4.1. We again perform additional experiments with different windows sizes (5, 8, 10, 13, 15, 18, 20, 23, 25, 28, and 30). We show the results of all windows sizes in figures while the final best performing model **win-13-L** is shown in tables. We also generate new segments for every window size parameter setting, but report only the best results here, obtained by **newSeg-V1-win-5** and **newSeg-V2-win-5**.

Table 1 shows the comparison between the baseline system and **win-13-L** in argument based evaluations. Our system outperforms the baseline system when identifying almost all semantic roles. The results for the basic semantic role types likes: ARG0, ARG1, ARG3, ARGM-LOC, ARGM-MNR, ARGM-MOD seem quite satisfactory, with the F_1 score typically above 75%. On the other hand, the system does not perform well when identifying semantic role ARG2 compared to the other semantic roles. It was to be expected knowing that the identification of ARG2 is still a running problem in NLP SRL systems. From the table 1, it is also clear that our system is far better than the baseline system in predicting circumstantial argument roles like ARGM-LOC, ARGM-MNR, and ARGM-MOD which occur far from the predicate, and our system is able to correctly identify them because of the *take-longest* text span selection approach.

Figures 4(a), 4(b), and 4(c) show the argEval evaluations across all windows sizes with the longest and the shortest text span selection. In general, our models outperform the baseline system in terms of recall and F_1 measure. However, the models from **win-10-L** to **win-30-L** exhibit lower precision scores than the baseline system, which indicates that by increasing the window size, we add more noise in the evidence that is used by the system to detect correct arguments and text spans. The figures also reveal that, as we increase the windows size, the recall scores increase while precision scores decrease. We may also notice that the system is better in identifying correct semantic roles using the take-shortest approach to text span selection (when compared to take-longest) with larger window sizes, since very frequent semantic roles like ARG0 and ARG1 are typically composed of only a few words. However, this is not the case when the system evaluates the predicate-semantic role pair, as shown in figures 5(d), 5(e), and 5(f). In this evaluation, **Win-13-L** outperforms all other models as well as the baseline system. The results are further displayed in table 1. The model outperforms the baseline system when predicting the semantic roles ARG1, ARG2, ARG3, ARGM-LOC with their predicates but could not beat the baseline results when predicting ARG0, ARGM-MNR, ARGM-MOD. When we provide the correct predicate to our model, then our model outperforms the baseline system for semantic roles ARG0, ARGM-MNR as shown for the corrPredArgEval evaluation in table 1. This indicates that our SRL system

Table 1. A comparison of results obtained using the baseline model and **Win-13-L** in three different evaluation protocols (argEval, predArgEval and corrPredArgEval)

	argEval			predArgEval			corrPredArgEval		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Baseline									
arg0	0.9889	0.7963	0.8822	0.7731	0.6225	0.6897	0.8644	0.6225	0.7238
arg1	0.9945	0.6805	0.8080	0.7479	0.5118	0.6077	0.8507	0.5118	0.6391
arg2	0.9929	0.3242	0.4888	0.6687	0.2184	0.3292	0.7386	0.2184	0.3371
arg3	0.9630	0.5219	0.6769	0.6498	0.3522	0.4568	0.7096	0.3522	0.4707
argm-loc	0.9506	0.4711	0.6300	0.5236	0.2595	0.3470	0.5819	0.2595	0.3589
argm-mnr	0.9457	0.4587	0.6178	0.5127	0.2487	0.3349	0.5746	0.2487	0.3471
argm-mod	0.9910	0.6640	0.7952	0.8533	0.5717	0.6847	0.9192	0.5717	0.7049
Macro average PRF	0.5430	0.2525	0.3447	0.3051	0.1543	0.2050	0.3383	0.1543	0.2120
Micro average PRF	0.9849	0.5803	0.7303	0.7204	0.4244	0.5342	0.8075	0.4244	0.5564
win-13-L									
arg0	0.9613	0.9023	0.9309	0.7067	0.6633	0.6843	0.8230	0.6659	0.7362
arg1	0.9932	0.7594	0.8607	0.7384	0.5645	0.6399	0.8647	0.5675	0.6853
arg2	0.9848	0.4226	0.5914	0.6001	0.2575	0.3604	0.6784	0.2582	0.3740
arg3	0.9084	0.6515	0.7588	0.5903	0.4234	0.4931	0.6472	0.4252	0.5132
argm-loc	0.8655	0.7674	0.8135	0.3789	0.3360	0.3562	0.4307	0.3405	0.3803
argm-mnr	0.9056	0.7078	0.7946	0.3738	0.2922	0.3280	0.4277	0.2939	0.3484
argm-mod	0.9865	0.6869	0.8098	0.8125	0.5657	0.6670	0.8849	0.5669	0.6910
Macro average PRF	0.5341	0.3365	0.4129	0.2949	0.1773	0.2215	0.3231	0.1780	0.2295
Micro average PRF	0.9581	0.7009	0.8096	0.6388	0.4673	0.5397	0.7363	0.4694	0.5733

Table 2. A comparison of results obtained using the baseline model and our models in predicate and all its semantic roles evaluation (predArgFullFrameEval)

	Precision	Recall	F_1
baseline	**0.2646**	**0.1865**	**0.2188**
win-5-L	**0.2452**	**0.1825**	**0.2093**
win-8-L	0.2344	0.1775	0.2020
win-10-L	0.2266	0.1724	0.1958
win-13-L	0.2176	0.1659	0.1883
win-15-L	0.2139	0.1636	0.1854
win-17-L	0.2096	0.1603	0.1817
win-20-L	0.2062	0.1593	0.1797
win-23-L	0.2045	0.1565	0.1773
win-25-L	0.2024	0.1547	0.1754
win-28-L	0.2023	0.1544	0.1751
win-30-L	0.2008	0.1532	0.1738
newSeg-V1-win-5	0.1874	0.1671	0.1767
newSeg-V2-win-5	0.1683	0.1755	0.1718

outputs different semantic roles according to the selected segment length, and selecting optimal segment length is essential for the overall performance of the system.

However, our models are unable to improve over the baseline system in the full frame evaluation in terms of precision, recall and F_1, although the results of **win-5-L** comes very close to the results of the baseline system, and is on a par with only a 0.4% lower recall score (not significant at $p < 0.005$ using a two-tailed t-test) and a 1.94% lower precision score (not significant at $p < 0.005$ using the same significance test). This evaluation is very strict since one missed or wrongly classified argument respectively results in a non-recognised or wrongly recognised frame. We have also investigated whether a small window size is better than larger window sizes in order to predict the full frame of a predicate. From figure 5(a), 5(b), and 5(c), it is visible that the take-longest approach to text span selection with smaller windows produces better results than the models relying on the take-shortest heuristic. We hope that our novel modeling principles may lead to further developments and new approaches to identifying predicates with their semantic roles without the need of sentence boundary detection, which is still a major problem for ASR-transcribed speech data [5].

6 Conclusion and Future Work

We have proposed a novel approach to identify PropBank-style semantic predicates and their semantic role arguments in speech data. The specific problem that we tackle in this paper concerns the absence of any sentence boundaries in transcribed speech data. We have shown that even with a very simple, but robust segmentation approach we attain results that are competitive or even better than state-of-the-art results on the OntoNotes speech data set. We have analysed different approaches to selecting correct predicates, arguments and their text spans.

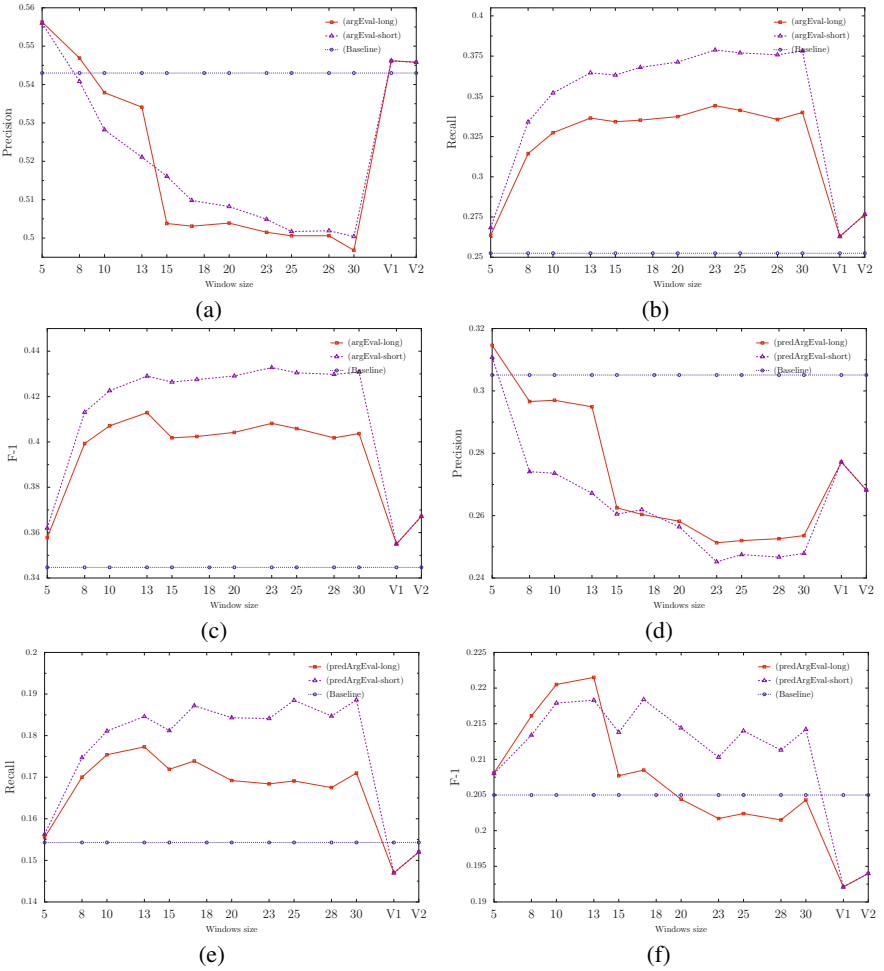

Fig. 4. Influence of windows size (when performing take-shortest and take-longest text span selection) on the overall results: (a) precision on argEval, (b) recall on argEval, (c) F_1 on argEval, (d) precision on predArgEval, (e) recall on predArgEval, and (f) F_1 on predArgEval. (In all figures, V1 and V2 are the best results obtained from **newSeg-V1-win-5** and **newSeg-V2-win-5** respectively.)

This work offers ample opportunities for further research. Currently, we do not employ any linguistic information in our models. The linguistic information will be exploited in future research in the form of language models and word embeddings trained on representative corpora, or in the form of shallow syntactic analyses of speech fragments (e.g., in the form of dependency information of phrases). We used an off-the-shelf SRL trained on written text data. We could investigate whether this SRL model could be transferred to speech data following the recent work from [18].

As another contribution we have built gold standard alignments between the predicates and arguments annotated in OntoNotes (which form the correct transcripts) and

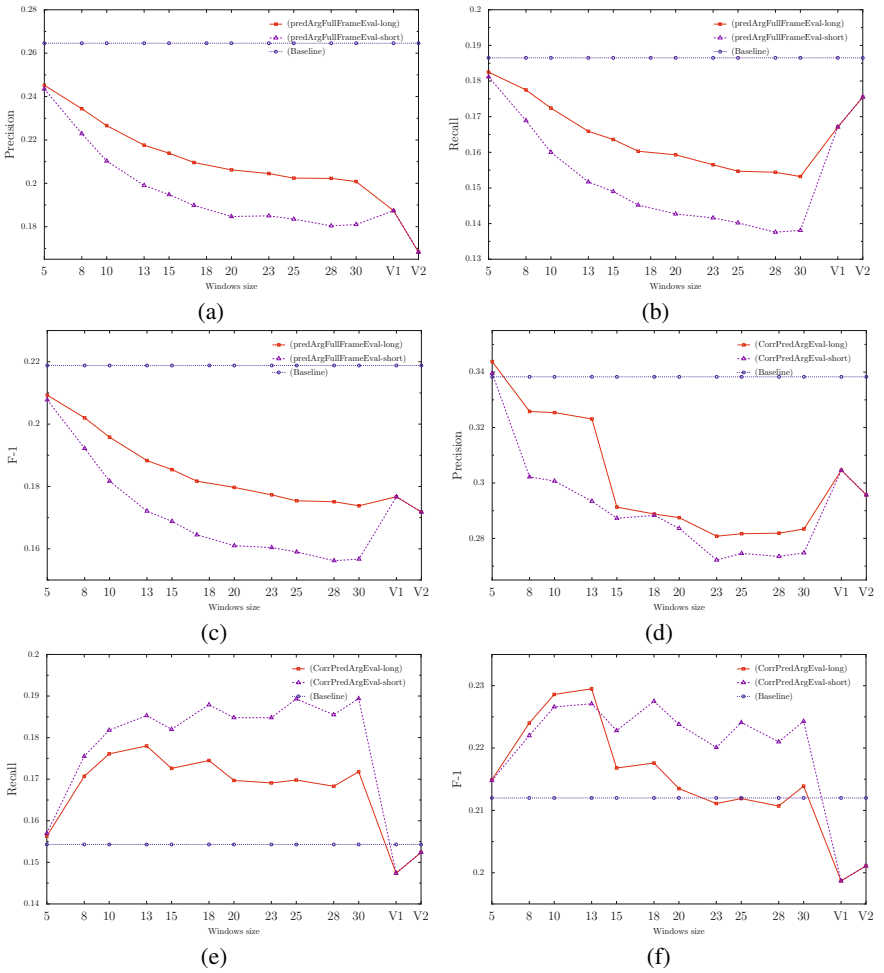

Fig. 5. Influence of windows size (when performing take-shortest and take-longest text span selection) on the overall results: (a) precision on predArgFullFrameEval, (b) recall on predArgFullFrameEval, (c) F_1 on predArgFullFrameEval, (d) precision on corrPredArgEval, (e) recall on corrPredArgEval, and (f) F_1 on corrPredArgEval. (In all figures, V1 and V2 are the best results obtained from **newSeg-V1-win-5** and **newSeg-V2-win-5** respectively.)

the original speech transcripts that were used in this and state-of-the-art research. This way we have produced an annotated corpus aligned to the original speech transcripts. We foresee that such a resource will be very valuable in future research on this topic.

References

1. Pradhan, S., Hacioglu, K., Ward, W., Martin, J.H., Jurafsky, D.: Semantic role chunking combining complementary syntactic views. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, CoNLL 2005, Stroudsburg, PA, USA, pp. 217–220 (2005)

2. Johansson, R., Nugues, P.: Dependency-based semantic role labeling of PropBank. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 69–78. Association for Computational Linguistics, Stroudsburg (2008)
3. Surdeanu, M., Turmo, J.: Semantic role labeling using complete syntactic analysis. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, CoNLL 2005, pp. 221–224. Association for Computational Linguistics, Stroudsburg (2005)
4. Punyakanok, V., Roth, D., Yih, W.: The importance of syntactic parsing and inference in seamantic role labeling. Comput. Linguist. 34, 257–287 (2008)
5. Favre, B., Bohnet, B., Hakkani-Tür, D.: Evaluation of semantic role labeling and dependency parsing of automatic speech recognition output. In: Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 5342–5345 (2010)
6. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: The 90% solution. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. NAACL-Short 2006, pp. 57–60. Association for Computational Linguistics, Stroudsburg (2006)
7. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Comput. Linguist. 28, 245–288 (2002)
8. Màrquez, L., Carreras, X., Litkowski, K.C., Stevenson, S.: Semantic role labeling: An introduction to the special issue. Comput. Linguist. 34, 145–159 (2008)
9. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. Comput. Linguist. 31(1), 71–106 (2005)
10. Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., Grishman, R.: The NomBank project: An interim report. In: Proceedings of the NAACL/HLT Workshop on Frontiers in Corpus Annotation (2004)
11. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, ACL 1998, vol. 1, pp. 86–90. Association for Computational Linguistics, Stroudsburg (1998)
12. Zhao, H., Chen, W., Kit, C., Zhou, G.: Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2009, Boulder, Colorado, USA, June 4, pp. 55–60 (2009)
13. Stenchikova, S., Hakkani-Tür, D., Tür, G.: QASR: question answering using semantic roles for speech interface. In: Proceedings of INTERSPEECH (2006)
14. Kolomiyets, O., Moens, M.F.: A survey on question answering technology from an information retrieval perspective. Inf. Sci. 181, 5412–5434 (2011)
15. Hüwel, S., Wrede, B.: Situated speech understanding for robust multi-modal human-robot communication (2006)
16. Huang, X., Baker, J., Reddy, R.: A historical perspective of speech recognition. Commun. ACM 57, 94–103 (2014)
17. Mohammad, S., Zhu, X., Martin, J.: Semantic role labeling of emotions in Tweets. In: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 32–41. Association for Computational Linguistics, Baltimore (2014)
18. Ngoc Thi Do, Q., Bethard, S., Moens, M.F.: Text mining for open domain semi-supervised semantic role labeling. In: Proceedings of the First International Workshop on Interactions Between Data Mining and Natural Language Processing, pp. 33–48 (2014)